Sisteme de Prelucrare Grafică

Curs nr. 1

Evoluţia graficii: Se pot distinge mai multe etape:

- grafica simplă care să fie printată;
- modele sau obiecte care trebuiau modelate grafic (obiecte abstracte şi fenomene);
- grafică interactivă – se modifică în timp real modelele construite;
- apar interfeţe grafice utilizator care permit realizare aplicaţiilor grafice complexe;
- dezvoltarea a două subdomenii:
    a) sinteza de imagini;
    b) analiza şi prelucrarea imaginilor.

a) Sinteza imaginilor constituie procesul de redare a unor obiecte reale sau imaginare, de obicei pornind de la un model. Sinteza de imagini a avut ca rezultat redarea fotorealistă a imaginilor (redare - rending). Prin redare fotorealistă se înţelege simularea fenomenelor de interacţiune a luminii cu diferite obiecte (reflexia + reflacţia).

b) Analiza şi prelucrarea imaginilor: reprezintă procesul invers celui de sinteză şi constă în prelucrarea imaginii sau reconstrucţia unui obiect pronind de la imagini ale sale. În acest subdomeniu, s-au dezvoltat 3 direcţii:
    a. îmbunătăţirea calităţii imaginii – *image enhancement* – constă în diminuarea sau eliminarea „zgomotului" din imagini (zgomot – pixeli suplimentari sau absenţi);
    b. detectarea şabloanelor standard (pattern detection & recognition) şi determinarea gradului de distorsiune în imagini 2D. (Ex: tehnica OCR – recunoaşterea optică a caracterelor după şablon).
    c. analiza de imagini şi computer vision (image analysis and computer vision) constă în determinarea obiectelor 2D/3D pornind de la imagini 2D (Ex: roboţii industriali disting obiectele, poziţia şi proprietăţile lor).

Iniţial sinteza era considerată o disciplină diferită de analiză şi prelucrare. În prezent cele două subdomenii constituie grafica pe calculator.

Concepte folosite în grafică

1. **Sistem grafic:** - ansamblu format din echipamente şi programe specializate în prelucrarea grafică. În funcţie de prelucrarea efectuată, sistemele grafice se grupează în două categorii:
    a. Sistem grafic de sinteza a imaginilor;
    b. Sistem grafic de analiză şi prelucrarea imaginilor.

Secvenţa de prelucrări efectuate într-un sistem de sinteză a imaginilor poate fi reprezentată schematic astfel:

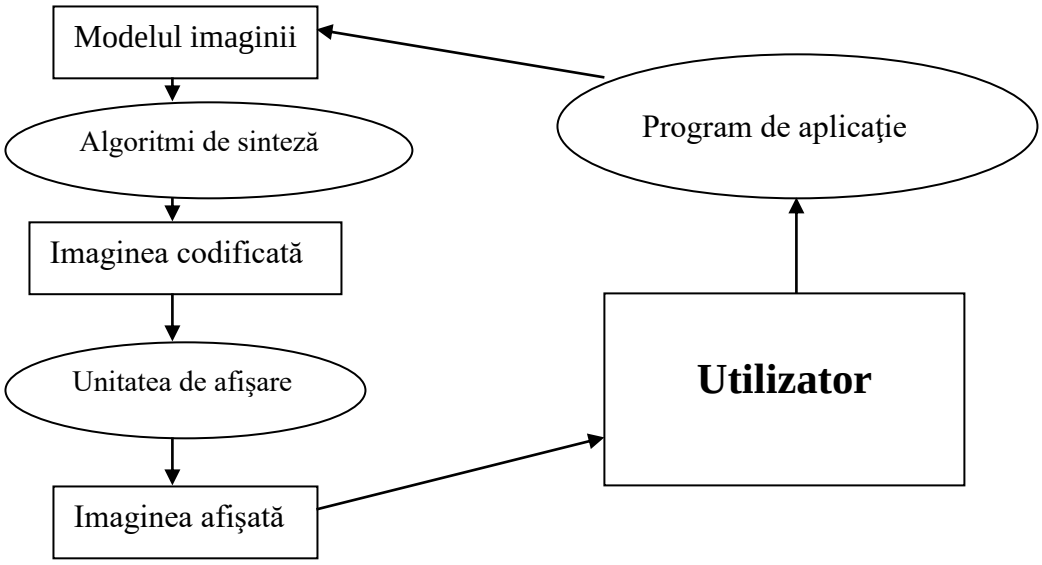


Fig. 1 – Sistem de sinteză a imaginii

În sistemele de analiză şi prelucrare se porneşte de la imaginea reală. Asupra ei se aplică:
- agoritmi de analiză $\rightarrow$modelul imaginii;
- algoritmi de prelucrare$\rightarrow$o imagine afişată pe ecran.

2. **Grafica interactivă.** Cadrul conceptual al graficii interactive costă din:
    1. programul aplicaţiei.
    2. sistemul grafic.
    3. modelul aplicaţiei.

Programul aplicaţie: creează, memorează şi preia date de la modelul aplicaţiei. Sistemul grafic produce imaginea la dispozitivul de afişare, el fiind realizat fie sub forma unor subrutine într-o bibliotecă grafică, fie cu dispozitive hardware (procesor de afişare) capabil să interpreteze comenzi de desenare.

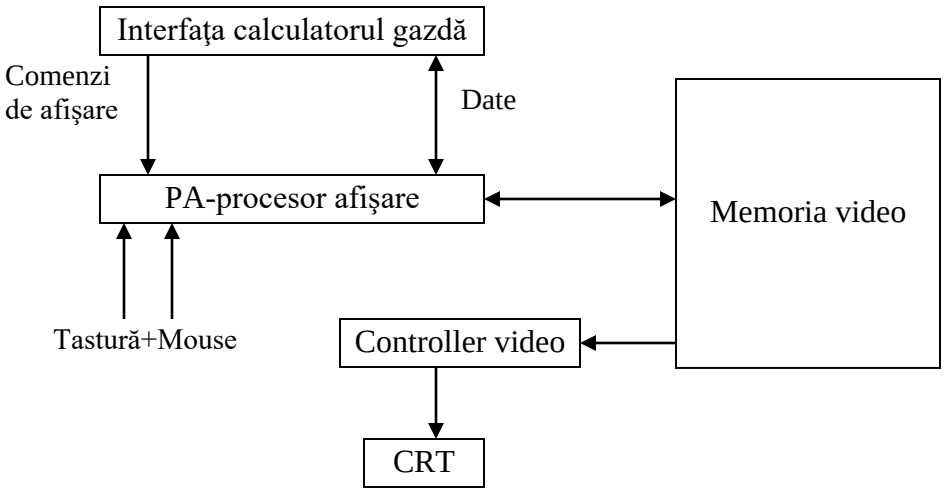
Modelul aplicaţiei conţine o descriere a obiectelor ce compun imaginea, obiecte care sunt descrise prin atribute geometrice (poziţie, dimensiune, orientare) şi atribute de afişare (culoare, tip linie, tip interior etc.)

3. **Tehnologii de afişare.** Există două tehnologii:
    a) Tehnologia vector
    b) Tehnologia raster

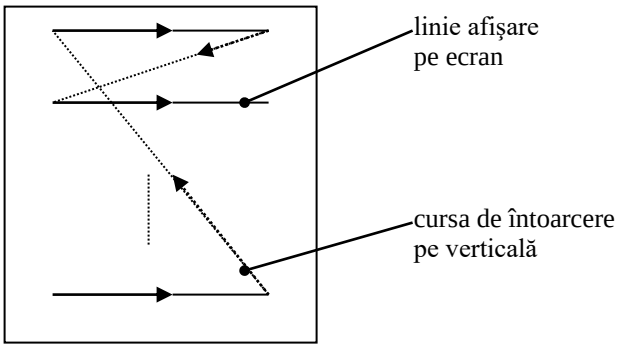
a) Tehnologia vector (linii) generarea imaginii se realizează prin trasarea de linii. Din cauza necesităţii de refresh periodic a ecranului imaginea trebuie generată ciclic.

Obs.: În memoria video nu avem imaginea, ci programul care construieşte imaginea ⇒ că imaginile nu pot fi complexe.

b) Tehnologia raster - tehnologia folosită în televiziune.
Memoria video nu conţine programul ci imaginea constituită din puncte (pixeli).
Structura sistemului este următoarea:



Procesul de afişare poate exista ca o componentă de soft inclusă în biblioteca grafică. În memoria video poate fi o zonă de memoria UCP (pentru staţii grafice de cost scăzut) sau poate fi o componentă hard (RIP – Raster Image Procesor) care acţionează ca un accelerator sau coprocesor în sistemul grafic. Imaginea este afişată de controllerul video (componentă hard), ea fiind alcătuită din linii orizontale. Sarcina controllerului este de a afişa linie cu linie, pe verticală, liniile orizontale.



Imaginea codificată în memoria video foloseşte :

- 1 bit/pixel – pentru monocrom (0- stins, 1- aprins)
- n biţi/pixel – pentru imagine color:
  n=8 - 256 de culori distincte pe ecran
  n=24 (fine colour) - 1 octet pentru fiecare din componentele primare RGB
  n=32 - 24 pentru RGB

- 8 biţi de control

n=96 - 64=2*31 pentru reprezentarea culorii în două buffere distincte: unul se afişează şi al doilea e folosit pentru generarea celei de-a doua instanţe a imaginii de afişat – double buffering

- 32 pentru algoritmii de determinare a poziţiei vizibile a obiectelor (pentru memorarea coordonatei z a unui punct)

Avantajele tehnologiei raster:

- independenţa ratei de refresh (reîmprospătare) de complexitatea imaginii de afişat
- posibilitatea de a reprezenta suprafeţe opace sau texturi

Dezavantaje:

- reprezentarea imaginii în memoria video sub formă de pixeli necesită algoritmi de conversie specifici (de exemplu algoritmul Bresenham pentru generarea de linii)
- reprezentarea aproximativă a primitivelor grafice (linii, cercuri etc.) prin puncte, rezultând un efect vizual nedorit, efect numit *aliasing* (acesta se datorează eşantionarii unei funcţii continue)

4. Portabilitate şi standarde grafice

Iniţial pachetele grafice erau dependente de dispozitiv şi scrise în limbaj low level pentru fiecare dispozitiv. Apoi s-au creat pachete independente de dispozitiv scrise în limbaje high, ceea ce a dus la portabilitatea aplicaţiilor, de unde a apărut şi necesitatea standardizarii unui sistem grafic. Prima specificaţie de SG3D a fost specificaţia CORE în 1977(SUA). În Europa în 1985 apare standardul GKS (Graphical Kernel System) – specificaţie a unui SG2D. În 1988 este extins la 3D. În 1988 în SUA apare un nou standard PHIGS (Programer's Hierarhical Interactive Graphics System).

În cadrul standardului PHIGS primitivele pot fi grupate în structuri care pot fi incluse ierarhic una în alta.
În GKS-3D se poate face gruparea primitivelor în structuri, dar acestea nu pot fi imbricate.

Ulterior, standardul PHIGS a fost extins la PHIGS+. Acesta conţine tehnici de redare pseudorealiste a scenelor 3D (simularea simplă a legilor opticii privind numai reflexia luminii între obiecte).
Au fost elaborate standarde pentru memorarea imaginii în fişiere:

- CGM – se foloseşte un limbaj de descriere a imaginii
- IGES – se combină în acelaşi fişier imagini şi text

Interfeţele grafice au devenit şi ele standarde grafice, utimul fiind OpenGL care asigură portabilitatea între diferite sisteme de operare (Unix, Windows).
În laborator – GLUT= OpenGL Utility Toolkit. În acest caz aplicaţiile scrise în C pot fi compilate şi executate pe o maşină Unix fără modificare.

Cap.1 Transformări grafice bidimensionale 2D

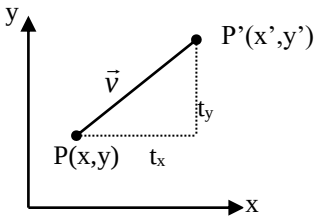
Există două interpretări ale unei transformări grafice a unui punct:

a) se efectuează transformări asupra coordonatelor unui punct, păstrând acelaşi sistem de coordonate. Această transformare se numeşte transformare geometrică (se foloseşte în grafică)

b) se efectuează transformări asupra sistemului de coordonate. Se foloseşte în exprimarea operaţiilor de vizualizare a obiectelor, care de obicei sunt definite într-un sistem de coordonate proprii, în timp ce afişarea se face într-un sistem de coordonate (SC) proprii suprafeţei.

1.1. Transformări geometrice 2D

a) Translaţia



$$\vec{v} = t_x * \vec{i} + t_y * \vec{j}$$

$$\boxed{\begin{aligned} x' &= x + t_x \\ y' &= y + t_y \end{aligned}} \qquad (1.1)$$

Obiectul e deplasat fără să modifice forma, dimensiunea, orientarea.

b) Scalarea

b.1 Scalarea faţă de origine

$$P(x, y) \rightarrow P'(x', y') \qquad \boxed{\begin{aligned} x' &= x * s_x \\ y' &= y * s_y \end{aligned}} \qquad (2)$$

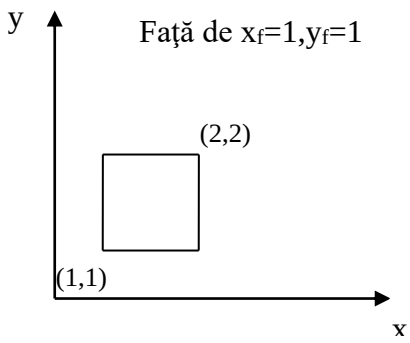
s_x, s_y – factori de scalare:

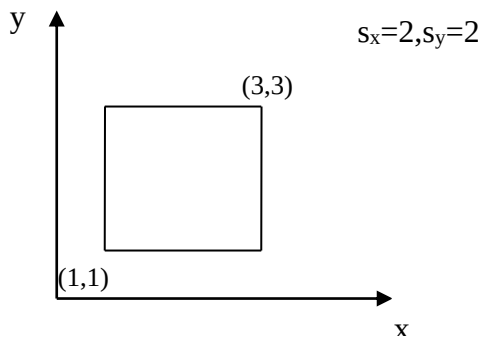
- $s_x = s_y$ – scalare uniformă $\Rightarrow$ obiectul este nedeformat
- $s_x \neq s_y$ – scalare neuniformă $\Rightarrow$ obiectul este deformat

b.2 Scalarea faţă de un punct oarecare din plan

$F(x_f, y_f)$ – punctul faţă de care se face scalarea

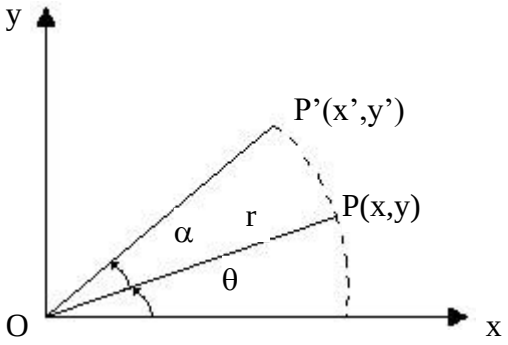
$$\begin{aligned} x' - x_f &= (x - x_f) * s_x \\ y' - y_f &= (y - y_f) * s_y \end{aligned} \quad \Rightarrow \quad \boxed{\begin{aligned} x' &= x * s_x + x_f - x_f * s_x \\ y' &= y * s_y + y_f - y_f * s_y \end{aligned}} \qquad (3)$$





c) Rotaţia

c.1 Rotaţia faţă de origine

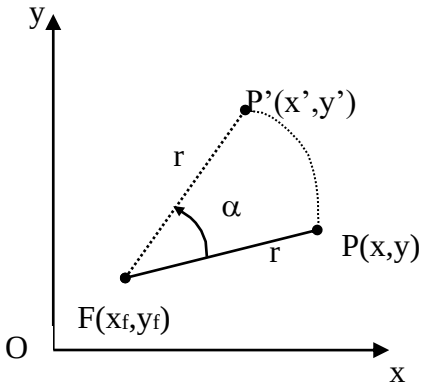


Unghiurile se consideră pozitive în sens trigonometric şi în sens invers.

$$\begin{cases} x = r\cos\theta \\ y = r\sin\theta \end{cases} \qquad \begin{cases} x' = r\cos(\theta+\alpha) = r\cos\theta\cos\alpha - r\sin\theta\sin\alpha \\ y' = r\sin(\theta+\alpha) = r\sin\theta\sin\alpha + r\cos\theta\sin\alpha \end{cases}$$

$$\boxed{\begin{cases} x' = r\cos\theta - y\sin\alpha \\ y' = r\sin\theta + y\cos\alpha \end{cases}} \qquad (4)$$

c.2 Rotaţia faţă de un punct oarecare fix (x_f,y_f)



$$\begin{cases} x' - x_f = (x - x_f)\cos\alpha - (y - y_f)\sin\alpha \\ y' - y_f = (x - x_f)\sin\alpha + (y - y_f)\cos\alpha \end{cases}$$

$$\boxed{\begin{cases} x' = x\cos\alpha - y\sin\alpha + x_f - x_f\cos\alpha + y_f\sin\alpha \\ y' = x\sin\alpha + y\cos\alpha + y_f - x_f\sin\alpha - y_f\cos\alpha \end{cases}} \qquad (5)$$

1.1.1. Formularea motivală a transformărilor

Coordonatele punctelor se pot reprezenta prin vectori de linie P=[x,y] sau vectori coloană $P^T=\begin{bmatrix} x \\ y \end{bmatrix}$. Dacă se folosesc vectori linie, atunci se efectuează înmulţire la dreapta cu matricea de transformare M, iar pentru vectori coloană înmulţirea se face la stânga cu M^T.

Obs: -În OpenGL se folosesc vectorii coloană

$(P*M)^T=M^T*P^T$ $\qquad M_L=M_C{}^T$

Vector linie Vector coloană $\qquad M_C=M_L{}^T$

În continuare utilizăm vectori coloană.
Dacă avem de efectuat mai multe transformări înseamnă că se efectuează compunerea de matrici.

Scalarea faţă de origine

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} \qquad (6) \qquad P'=S*P$$

Rotaţia faţă de origine

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} \qquad (7) \qquad P'=R*P$$

Obs: Pentru vectori linie $[x', y'] = [x, y] \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$

Translaţia

Nu se poate exprima ca un produs matricial, de aceea se foloseşte o adunare de vectori.

$P' = M + P$ unde $M = \begin{bmatrix} s_x \\ s_y \end{bmatrix}$, M-vector

Coordonate omogene

Deoarece pentru translaţia în coordonate carteziene nu există o matrice de transformare, pentru a uniformiza notaţiile se folosesc coordonate omogene. Acest sistem de coordonate permite reprezentarea tuturor transformărilor ca produs de matrici.

Unui punct 2D din plan (x,y) îi corespunde, în coordonate omogene un triplet (x_w,y_w,w), în care $\begin{cases} x = \frac{x_w}{w} \\ y = \frac{y_w}{w} \end{cases}$.

Obs: Dacă w=0, atunci pentru [a,b,0], punctul se consideră la infinit pe dreapta a*y-b*x=0.

Ex:

1) [1,0,0] – punctul de la infinit pe axa x pozitivă
[1,1,0] – punctul de la infinit în direcţia [1,1]

2) Punctul P(2,1) în coordonate plane carteziene, devine în coordonate omogene (2,1,1) sau (4,2,2) etc.

În grafică se utilizează w=1, ceea ce înseamnă că $P_{coord.carteziene}(x, y) \rightarrow P_{coord.omogene}(x, y, 1)$.

Dacă în urma unor transformări se obţine $w \neq 1$, se efectuează împărţirea cu w. În general, transformările geometrice primitive conservă valoarea lui w şi deci dacă se alege w=1 împărţirea nu mai este necesară.

Cele trei transformări elementare examinate anterior, într-un sistem omogen de coordonate se exprimă asfel:

Translaţia

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (8)$$

$$P' = T * P \qquad P = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad P' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Scalarea faţă de origine

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (9) \qquad P' = S * P$$

Rotaţia faţă de origine

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (10) \qquad P' = R * P$$

Obs: La aplicarea a două transformări asupra unui punct trebuie avut în vedere că produsul matricial nu e comutativ. Excepţie fac transformările de acelaşi tip.

Expresiile matematice ale scalării şi rotaţiei faţă de un punct oarecare se pot obţine prin compunerea următoarelor transformări:

1. – translaţia prin care punctul fix al transformării ajunge în origine
2. – scalarea/rotaţia faţă de origine
3. – translaţia inversă celei de la punctul 1.

Fie $F(x_f,y_f)$ punctul fix al transformării, α unghiul de rotaţie şi s_x,s_y factorii de scalare.

Scalarea faţă de un punct fix (x_f,y_f) $P' = T(x_f, y_f) * S(s_x, s_y) * T(-x_f,-y_f) * P$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (11)$$

Obs:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} s_x & 0 & -s_x * x_f \\ 0 & s_y & -s_y * y_f \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & -s_x * x_f + x_f \\ 0 & s_y & -s_y * y_f + y_f \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x * s_x + x_f - s_x * x_f \\ y * s_y + y_f - s_y * y_f \\ 1 \end{bmatrix}$$

adică am obţinut relaţiile (3).

Rotaţia faţă de un punct fix (x_f,y_f) $P' = T(x_f, y_f) * R(\alpha) * T(-x_f,-y_f) * P$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (12)$$

Temă de laborator:

a. Să se rotească de N ori un pătrat construit din linii în jurul centrului ecranului, cu un unghi specificat.

b. Să se efectueze translaţia unui pătrat asfel încât centrul său să parcurgă circumferinţa unui arc cu centrul în centrul ecranului.

1.1.2. Transformări geometrice inverse

Fiecare dintre transformările geometrice elementare are o inversă, adică o transformare care exprimă operaţia opusă celei corespunzătoare transformării.
Se pot verifica uşor următoarele relaţii:

$$[T(t_x,t_y)]^{-1}=T[(-t_x,-t_y)] \qquad (13)$$

$$[S(s_x,s_y)]^{-1}=S[(1/s_x,1/s_y)] \qquad (14)$$

$$[R(\alpha)]^{-1}=R(-\alpha) \qquad (15)$$

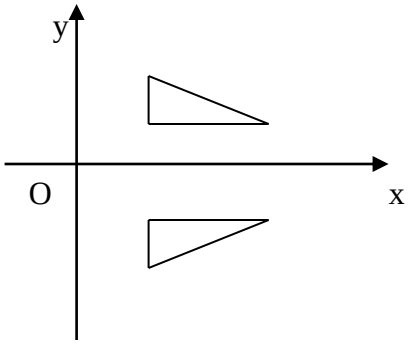
De exemplu, pentru relaţia (13) avem:

$$T(tx,ty) * T(-tx,-ty) = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -tx \\ 0 & 1 & -ty \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad q.e.d.$$

1.1.3. Alte transformări simple

a) Oglidirea (Reflexia)

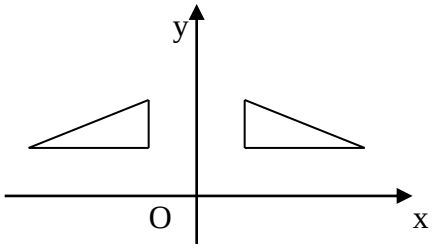
– Faţă de axa x



$$\begin{cases} x' = x \\ y' = -y \end{cases}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (16) \qquad O_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

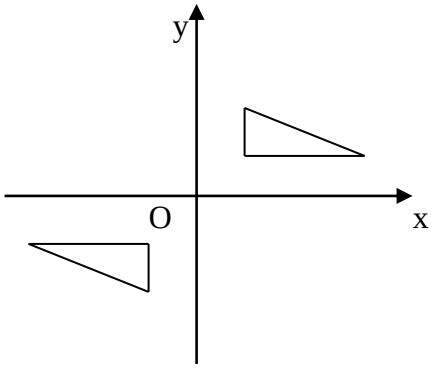
– Faţă de axa y



$$\begin{cases} x' = -x \\ y' = y \end{cases}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (17) \qquad O_y = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

– Faţă de origine

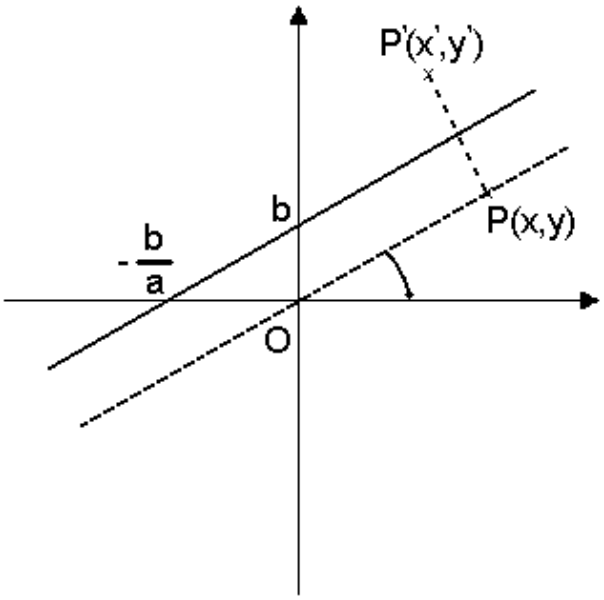


$$\begin{cases} x' = -x \\ y' = -y \end{cases}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (18) \qquad O_o = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

– Oglindirea faţă de o dreaptă oarecare

Fie dreapta z=a∗x+b



Se poate exprima oglindirea faţă de o dreaptă y=a∗x+b, ca o transformare compusă din următoarele transformări elementare:

1. o translaţie astfel încât o dreaptă să treacă prin origine T(0,-b)
2. o rotaţie a dreptei pentru a o alinia cu una din axele de coordonate R(-arctg(α)) (de exemplu cu axa Ox)
3. oglidirea faţă de axa pe care a fost suprapusă Ox
4. rotaţie inversă celei de la punctul 2, R(arctg(α))
5. translaţie inversă celei de la punctul 1, T(0,b)

Matricea transformării este:

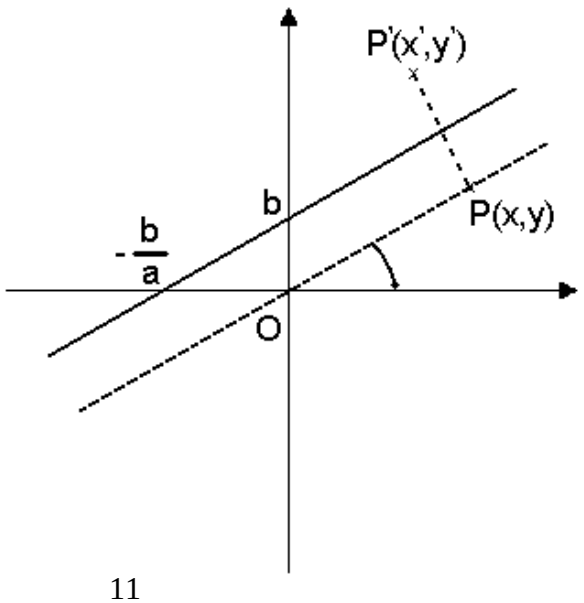
M=T(0,b)∗R(arctg(α))∗Ox∗R(-arctg(α))∗T(0,-b), deci | P'=M∗P |

Exemplu: y=x-2

a=1, b=-2

arctg1=45^0

$$P = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$



$$M=\begin{bmatrix}1&0&0\\0&1&-2\\0&0&1\end{bmatrix}*\begin{bmatrix}\frac{\sqrt{2}}{2}&-\frac{\sqrt{2}}{2}&0\\\frac{\sqrt{2}}{2}&\frac{\sqrt{2}}{2}&0\\0&0&1\end{bmatrix}*\begin{bmatrix}1&0&0\\0&-1&0\\0&0&1\end{bmatrix}*\begin{bmatrix}\frac{\sqrt{2}}{2}&\frac{\sqrt{2}}{2}&0\\-\frac{\sqrt{2}}{2}&\frac{\sqrt{2}}{2}&0\\0&0&1\end{bmatrix}*\begin{bmatrix}1&0&0\\0&1&2\\0&0&1\end{bmatrix}=$$

$$=\begin{bmatrix}1&0&0\\0&1&-2\\0&0&1\end{bmatrix}*\begin{bmatrix}\frac{\sqrt{2}}{2}&-\frac{\sqrt{2}}{2}&0\\\frac{\sqrt{2}}{2}&\frac{\sqrt{2}}{2}&0\\0&0&1\end{bmatrix}*\begin{bmatrix}1&0&0\\0&-1&0\\0&0&1\end{bmatrix}*\begin{bmatrix}\frac{\sqrt{2}}{2}&\frac{\sqrt{2}}{2}&\sqrt{2}\\-\frac{\sqrt{2}}{2}&\frac{\sqrt{2}}{2}&\sqrt{2}\\0&0&1\end{bmatrix}=$$

$$=\begin{bmatrix}1&0&0\\0&1&-2\\0&0&1\end{bmatrix}*\begin{bmatrix}\frac{\sqrt{2}}{2}&-\frac{\sqrt{2}}{2}&0\\\frac{\sqrt{2}}{2}&\frac{\sqrt{2}}{2}&0\\0&0&1\end{bmatrix}*\begin{bmatrix}\frac{\sqrt{2}}{2}&\frac{\sqrt{2}}{2}&\sqrt{2}\\\frac{\sqrt{2}}{2}&-\frac{\sqrt{2}}{2}&-\sqrt{2}\\0&0&1\end{bmatrix}=$$

$$=\begin{bmatrix}1&0&0\\0&1&-2\\0&0&1\end{bmatrix}*\begin{bmatrix}0&1&2\\1&0&0\\0&0&1\end{bmatrix}=\begin{bmatrix}0&1&2\\1&0&-2\\0&0&1\end{bmatrix}$$

$$\Rightarrow\begin{bmatrix}x'\\y'\\1\end{bmatrix}=\begin{bmatrix}0&1&2\\1&0&-2\\0&0&1\end{bmatrix}*\begin{bmatrix}1\\2\\1\end{bmatrix}=\begin{bmatrix}4\\-1\\1\end{bmatrix}\qquad\Rightarrow\begin{cases}x'=4\\y'=-1\end{cases}$$

Obs:

1) Ordinea de acţionare a matricilor care compun transformarea este de la dreapta la stânga în cazul vectorilor coloană (cazul considerat) şi de la stânga la dreapta în cazul vectorilor linie.

2) Pentru două transformări de acelaşi tip avem:
   $T_1*T_2=T_2*T_1$
   $R_1*R_2=R_2*R_1$
   $S_1*S_2=S_2*S_1$

Dacă $s_x=s_y \Rightarrow S*R=R*S$

3) Pentru mai multe transformări în secvenţă, ordinea este, de regulă, următoarea: scalarea se aplică înaintea rotaţiei.